

ASG-Assure™ User's Guide

Version 1.0

Publication Number: TCC0200-10

Publication Date: April 1999

The information contained herein is the confidential and proprietary information of Allen Systems Group, Inc. Unauthorized use of this information and disclosure to third parties is expressly prohibited. This technical publication may not be reproduced in whole or in part, by any means, without the express written consent of Allen Systems Group, Inc.

© 2002 Allen Systems Group, Inc. All rights reserved.

All names and products contained herein are the trademarks or registered trademarks of their respective holders.

ASG Documentation/Product Enhancement Fax Form

Please FAX comments regarding ASG products and/or documentation to (941) 263-3692.

Company Name	Telephone Number	Site ID	Contact name

Product Name/Publication	Version #	Publication Date
Product:		
Publication:		
Tape VOLSER:		

Enhancement Request:

ASG Support Numbers

ASG provides support throughout the world to resolve questions or problems regarding installation, operation, or use of our products. We provide all levels of support during normal business hours and emergency support during non-business hours. To expedite response time, please follow these procedures.

Please have this information ready:

- Product name, version number, and release number
- List of any fixes currently applied
- Any alphanumeric error codes or messages written precisely or displayed
- A description of the specific steps that immediately preceded the problem
- The severity code (ASG Support uses an escalated severity system to prioritize service to our clients. The severity codes and their meanings are listed below.)

If You Receive a Voice Mail Message:

- 1 Follow the instructions to report a production-down or critical problem.
- 2 Leave a detailed message including your name and phone number. A Support representative will be paged and will return your call as soon as possible.
- 3 Please have the information described above ready for when you are contacted by the Support representative.

Severity Codes and Expected Support Response Times

Severity	Meaning	Expected Support Response Time
1	Production down, critical situation	Within 30 minutes
2	Major component of product disabled	Within 2 hours
3	Problem with the product, but customer has work-around solution	Within 4 hours
4	"How-to" questions and enhancement requests	Within 4 hours

ASG provides software products that run in a number of third-party vendor environments. Support for all non-ASG products is the responsibility of the respective vendor. In the event a vendor discontinues support for a hardware and/or software product, ASG cannot be held responsible for problems arising from the use of that unsupported version.

Business Hours Support

Your Location	Phone	Fax	E-mail
United States and Canada	800.354.3578 1.941.435.2201 Secondary Numbers: 800.227.7774 800.525.7775	941.263.2883	support@asg.com
Australia	61.2.9460.0411	61.2.9460.0280	support.au@asg.com
England	44.1727.736305	44.1727.812018	support.uk@asg.com
France	33.141.028590	33.141.028589	support.fr@asg.com
Germany	49.89.45716.300	49.89.45716.400	support.de@asg.com
Singapore	65.224.3080	65.224.8516	support.sg@asg.com
All other countries:	1.941.435.2201		support@asg.com

Non-Business Hours - Emergency Support

Your Location	Phone	Your Location	Phone
United States and Canada	800.354.3578 1.941.435.2201 Secondary Numbers: 800.227.7774 800.525.7775 Fax: 941.263.2883		
Asia	011.65.224.3080	Japan/Telecom	0041.800.9932.5536
Australia	0011.800.9932.5536	New Zealand	00.800.9932.5536
Denmark	00.800.9932.5536	South Korea	001.800.9932.5536
France	00.800.9932.5536	Sweden/Telia	009.800.9932.5536
Germany	00.800.9932.5536	Switzerland	00.800.9932.5536
Hong Kong	001.800.9932.5536	Thailand	001.800.9932.5536
Ireland	00.800.9932.5536	United Kingdom	00.800.9932.5536
Israel/Bezeq	014.800.9932.5536		
Japan/IDC	0061.800.9932.5536	All other countries	1.941.435.2201

ASG Web Site

Visit <http://www.asg.com>, ASG's World Wide Web site.

Submit all product and documentation suggestions to ASG's product management team at <http://www.asg.com/products/suggestions.asp>

If you do not have access to the web, FAX your suggestions to product management at (941) 263-3692. Please include your name, company, work phone, e-mail ID, and the name of the ASG product you are using. For documentation suggestions include the publication number located on the publication's front cover.

Contents

Preface	iii
About this Publication	iii
Related Publications	iii
Publication Conventions	iv
Requesting Publication Changes	iv
1 Introduction	1
Features	2
Uses	4
Benefits	4
2 Installation	5
Technical Requirements	5
Hardware and Software Requirements	5
File and DASD Requirements	5
Assure's Installation Cartridge	6
Installing Assure	6
3 Creating the Assure How-changed Files	11
Preparing ASG-TRACER	11
Editing the @TE0BREF and @TG0BRRF JCL Jobstreams	11
Editing the //DBTINPUT Parameter File	12
Result Files	13
4 Using the Assure Summary Reports	15
TCRPSUMM—Processing Summary Report	15
TCRPCOMP—Comparison Summary Report	16
Combining the Process Reports	17
TCRFSUMM—Element Summary Report	21
TCRFCOMP—Element Comparison Summary	22
Combining the Field Reports	22
Appendix A	
//TCRSUMRY File	27
Appendix B	
//SUPCRECL File	29
Appendix C	

//SUPCFLDL File 39

Appendix D
Year 2000 Regression Testing 43

Index 45

Preface

This *ASG-Assure User's Guide* provides detailed instructions on how to install and use ASG-Assure (herein called Assure). Assure enables you to automate the comparison of application program updates made to your CA-IDMS database before and after you implement coding changes. ASG-TRACER extracts information from the CA-IDMS journal files whenever a program updates the database. Assure uses this information as input for comparison.

ASG welcomes your comments, as a preferred or prospective customer, on this publication or on the Assure product.

About this Publication

The *ASG-Assure User's Guide* consists of these chapters:

- Chapter 1, "Introduction," introduces the Assure product.
- Chapter 2, "Installation," describes how to install Assure.
- Chapter 3, "Creating the Assure How-changed Files," describes how to create the user exits in ASG-TRACER.
- Chapter 4, "Using the Assure Summary Reports," describes how to create and use the summary comparison reports.

Related Publications

The complete documentation library for ASG-Assure consists of these publications (where *nn* represents the version number):

- *ASG-Assure User's Guide* (TCC0200-*nn*) provides detailed instructions on how to install and use ASG-ASSURE.
- *ASG-TRACER User's Guide* (TCR0200-*nn*) provides a detailed explanation of how to install, invoke, and use the ASG-TRACER product.

Publication Conventions

Allen Systems Group technical publications use these conventions:

Convention	Represents
ALL CAPITALS	Directory, path, file, dataset, member, database, program, command, and parameter names.
Initial Capitals on Each Word	Window, field, field group, check box, button, panel (or screen), option names, and names of keys. A plus sign (+) is inserted for key combinations (e.g., Alt+Tab).
<i>lowercase italic monospace</i>	Information that you provide according to your particular situation. For example, you would replace <i>filename</i> with the actual name of the file.
Monospace	Characters you must type exactly as they are shown. Code, JCL, file listings, or command/statement syntax. Also used for denoting brief examples in a paragraph.

Requesting Publication Changes

Customers and other ASG departments can use a Documentation Correction/Enhancement Request Form to request corrections, updates, and enhancements to publications. The form is included in the front matter of each publication. Forms are also available from the Vice President of Technical Publications.

The Vice President of Technical Publications evaluates requests for documentation changes.

1

Introduction

As an Assure customer, you must execute these two controlled test runs of the programs that have been changed:

The baseline run. In the baseline run, the original (unchanged) version of the program(s) is used.

The test version run. In the test version run, the new (changed) version of the program(s) is used.

ASG-TRACER and Assure process the archive journal files resulting from each of the test runs to produce three files. These files can be input to three runs of a standard compare utility. Assure produces files that are designed to be processed by any standard compare program, such as the SUPERC program provided by IBM (see the *Interactive System Productivity Facility/Program Development Facility ISPF/PDF Guide and Reference* for detailed information regarding the SUPERC program).

The files contain varying amounts of detail information about the updates that each test run produced:

The SUMMARY file. This file contains counts of the number of record occurrences of each record type that was updated, broken down by the program that produced the update and the type of updated verb.

Examples:

For Record-type=CUSTOMER, Program=ABC, Verb=Store, Count=1 record

For Record-type=CUSTOMER, Program=ABC, Verb=MODIFY, Count=2 records

The RECORD-LEVEL file. This file contains at least one record for each record occurrence that was changed during the test run. When the updating verb is a MODIFY, then there are two records in this file—one that shows some of the record's data as it appeared BEFORE the change and the other as it appeared AFTER the change.

Examples:

Record=Customer, Program=ABC, Verb=STORE, Key=Doe, John, Image=AFTER

Record=Customer, Program=ABC, Verb=MODIFY, Key=Doe, Jane, Image=BEFORE

Record=Customer, Program=ABC, Verb=MODIFY, Key=Smith, Jane, Image=AFTER

The FIELD-LEVEL file. This file contains at least one record for each FIELD that changed during the test run. When the update verb is either a STORE or an ERASE, then each field in the record occurrence produces a record occurrence in this file showing the name of the field and the value of the data in the field. When the update verb is a MODIFY, then two records are written to this file for each changed FIELD in the MODIFIED record occurrence.

Examples:

- Record occurrences are associated with a single STORE CUSTOMER verb (one record in the Field-Level file for each RECORD-ELEMENT in the CUSTOMER record):

Record=Customer, Program=ABC, Verb=STORE, Field=CUSTOMER-NAME, Value=Doe, John, Image=AFTER

Record=Customer, Program=ABC, Verb=STORE, Field=CUSTOMER-NBR, Value=123456789, Image=AFTER

Record=Customer, Program=ABC, Verb=STORE, Field=LAST-BILLED-DATE, Value=02/17/99, Image=AFTER

- Record occurrences are associated with a single MODIFY CUSTOMER verb (here, the only field in the record that was changed is CUSTOMER-NAME):

Record=Customer, Program=ABC, Verb=MODIFY, Field=CUSTOMER-NAME, Value=Doe, Jane, Image=BEFORE

Record=Customer, Program=ABC, Verb=MODIFY, Field=CUSTOMER-NAME, Value=Smith, Jane, Image=AFTER

Note:

When the updating verb is either a CONNECT or a DISCONNECT Assure includes the SET-NAME that the record was connected to, and the OWNER-DBKEY if the set has been defined in the SCHEMA with owner pointers maintained.

By producing this level of detail to be processed by a comparison program, Assure allows you to compare the specifics of how each record occurrence is changed. This helps reduce problems that could arise if only a sampling of the changed database records was compared manually.

In addition to using a Standard Change compare utility, Assure also includes programs to summarize the detail information and run a "pre-compare" process to minimize the amount of data that you need to view.

Features

Assure is an extension of ASG-TRACER. In order to use Assure's features, you should have ASG-TRACER installed and functioning.

Assure is made up of two parts:

How-changed files. The first component of Assure builds files that show how the database was changed. In this publication, we refer to these files as *how-changed files*. These are the how-changed files:

- //TCRSUMRY
- //SUPCRECL
- //SUPCFLDL

You can use these files as input to the comparison utility to perform comparisons on your data. The comparison utility will provide immediate results; however, the output may be unmanageable due to its size.

Summaries. The second component of Assure takes the detailed information from the how-changed files and summarizes it. This summary reduces the amount of data input to the compare process. You can view your comparison data online or in summarized reports. The summary reports automatically drop components that are obviously different. This allows you to clearly identify processing and field differences.

Assure's components work together to:

- Produce record and field level detail records from IDMS journal output.

Assure is a group of programs that take journal output from IDMS and manipulate them to produce record and field level how-changed files.

- Read detail records and create summary reports.

Programs in Assure read the resulting how-changed files and create different summary reports. You can use the summary reports to determine how significant the database changes are between the baseline and test-version runs before you commit to running the standard compare utility [i.e., comparing how the programs change the database as they currently run (baseline) versus how they change the programs after they have been modified (test-version run)]. In cases where your programs modify the database in a significantly different manner, the standard compare facilities could produce large listings of differences.

- Compare summary data and create output parameters for a SORT.

The summary programs produce a machine readable interim file Assure uses as input to a comparison program. The comparison program compares the summary data and provides a high level report of the differences between the summary files. Once the differences are determined, the Assure programs output parameters that are used as input to a SORT. The SORT enables the selection of records that will be passed to the final comparison process.

- Enable the selection of records that will be passed to the final comparison process.

You can use the detail records in both record and element levels as input to the comparison utility. The comparison utility finds the differences in sequence and content of the detail records. The comparison utility can result in large amounts of data that are almost impossible to interpret.

However, if you combine the runs from each phase of the summary process, you can select for comparison the matching processing or the processing that is not similar. This gives you flexibility in choosing what you view. It also allows you to have a high level of confidence that the programs are producing the same results before even looking at the output data.

Uses

You can use Assure in Year 2000 compliance regression testing and as a final sign-off stage for many quality assurance procedures. (See Appendix D, "Year 2000 Regression Testing" on page 43 for detailed steps on Y2K regression testing.)

Assure helps your shop become more "assured" that the modifications you made to your software have accomplished your desired objective—that unintended changes to logic have not occurred. Assure facilitates regression testing.

Assure produces formatted files that you can use as input into different file comparison programs (e.g., IBM's SUPERC). ASG-TRACER and Assure produce files designed to be used with file comparison. The DDNAMES (in the provided JCL) are named so that the first 4 character positions are SUPC. This naming convention makes it easier to know which files produced by ASG-TRACER and Assure can be input into a file comparison program.

Note: _____

Even though the name suggests SUPERC, any file comparison program should work, as long as it allows specification of only certain columns for comparison.

Benefits

Assure offers your organization these benefits:

Journal comparison. Assure provides you with the ability to perform record-level and element-level comparisons between test processing journals. This allows you to quickly determine if the processing being performed is the same or even close to the same.

Straightforward reports. Assure allows you to establish differences between test processing journals quickly without having to wade through huge piles of paper.

Standard utilities. Assure uses standard utilities (e.g., SUPERC and SORT) to perform comparison functions. These standard utilities allow you to perform comparison functions easily on any site.

2

Installation

This chapter provides installation information for Assure and contains these sections:

Topic	Page
Technical Requirements	5
Assure's Installation Cartridge	6
Installing Assure	6

Technical Requirements

Hardware and Software Requirements

Assure runs on any IBM mainframe or compatible CPU. ASG-TRACER must be installed and running under a MVS/SP, MVS/XA, MSP/EX or MVS/ESA environment.

File and DASD Requirements

Assure consists of these files:

File Name	Description
YOUR.TCR.ASR.Rxxx.JCLLIB	JCL library with compile and run-time JCL for Assure.
YOUR.TCR.ASR.Rxxx.LOADLIB	Assure load library containing run-time programs.

Note: _____

The file names shown in this table contain the high-level qualifier *YOUR*. Replace this qualifier with a high-level qualifier for use at your site.

Direct Access Storage Device (DASD) requirements for your Assure files is five cylinders. DASD requirements for your intermediate datasets are dependent on the size and complexity of the transaction tests.

Assure's Installation Cartridge

ASG ships Assure in a non-compressed format on a cartridge or a 6250 bits per inch (bpi) tape. The installation cartridge contains these datasets:

Tape File Name	Tape File Number	File Description
TCR.ASR.Rxxx.JCLLIB	1	Assure JCL Library
TCR.ASR.Rxxx.LOADLIB	2	Assure Product Loadlib
TCR.ASR.Rxxx.UNLOAD	3	Assure tape unload JCL

Note:

When unloading tape file number 3, use the IEBGENER utility shown in Figure 1 on page 6. Do not use IEBCOPY.

Installing Assure

To install Assure, complete these steps:

1. Unload and customize the JCL in tape file number 3. You will use this JCL to read and unload the JCL from the Assure installation cartridge.
 - 1.1 To unload tape file number 3, create the IEBGENER job shown in Figure 1 on page 6.

```
//JOB CARD GOES HERE
// *
//INST      EXEC  PGM=IEBGENER
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD DSN=TCR.ASR.Rxxx.UNLOAD,DISP=OLD,LABEL=(3,SL),
//              VOL=SER=XXXXXX,UNIT=TAPE
//SYSUT2 DD DSN=yourname.UNLOAD(JCL),DISP=(,CATLG,DELETE),
//              DCB=(RECFM=FB,LRECL=80,BLKSIZE=3120),
//              SPACE=(TRK,(1,1,1),RLSE),UNIT=SYSDA,VOL=SER=volser
//SYSIN DD DUMMY
// *
```

Figure 1

- 1.2 To customize the JCL in Figure 1 on page 6, modify the job card and change these parameters using the appropriate values for your site:

Parameter	Description
<i>ASRxxx</i>	The volser of your Assure installation tape. Where <i>xxx</i> is equal to the release level of Assure.
<i>TAPE</i>	The name of your tape device.
<i>yourname</i>	The high level qualifier that your organization has chosen for Assure datasets.
<i>SYSDA</i>	The name of the appropriate DASD device.
<i>volser</i>	The volser of your DASD device.

2. Customize and run the JCL in tape file number 3 (Figure 2 on page 7) to read and unload the files from the Assure installation cartridge.

```
//JOBNAME JOB (ACCT),'TCR INSTALL',REGION=6M,
//          MSGCLASS=X,CLASS=A,NOTIFY=USERID
// *
//COPY1 EXEC PGM=IEBCOPY
//SYSPRINT DD SYSOUT=*
//TAPE01 DD DSN=TCR.ASR.Rxxx.JCLLIB,DISP=OLD,VOL=SER=( TCRXXX ),
//          UNIT=TAPE,LABEL=(1,SL)
//DISK01 DD DSN=YOUR.TCR.ASR.Rxxx.JCLLIB,DISP=(NEW,CATLG,DELETE),
//          UNIT=SYSDA,VOL=SER=XXXXXX,SPACE=(CYL,(1,2,10))
//TAPE02 DD DSN=TCR.ASR.Rxxx.LOADLIB,DISP=OLD,VOL=REF=*.TAPE01,
//          UNIT=TAPE,LABEL=(2,SL)
//DISK02 DD DSN=YOUR.TCR.ASR.Rxxx.LOADLIB,DISP=(NEW,CATLG,DELETE),
//          UNIT=SYSDA,VOL=SER=XXXXXX,SPACE=(CYL,(2,2,10))
//SYSUT3 DD UNIT=SYSDA,SPACE=(CYL,(3,3))
//SYSUT4 DD UNIT=SYSDA,SPACE=(CYL,(3,3))
//SYSIN DD *
COPY I=TAPE01,O=DISK01
COPY I=TAPE02,O=DISK02
```

Figure 2

- 2.1 To customize the JCL in Figure 2 on page 7, modify the job card and change these parameters using the appropriate values for your site:

Parameter	Description
<i>ASRxxx</i>	The volser of the Assure installation tape. Where <i>xxx</i> is equal to the release level of Assure.
<i>TAPE</i>	The name of your tape unit.
<i>YOUR.TCR.ASR.Rxxx.JCLLIB</i>	The name that your site has selected for the Assure JCL library.

Parameter	Description
<i>SYSDA</i>	The name or parameter value specifying the appropriate DASD unit.
<i>XXXXXXX</i>	The name or parameter value specifying the appropriate volser of your DASD device.
<i>YOUR.TCR.ASR.Rxxx. . .</i>	The names of the site selected libraries for your Assure datasets.

- 2.2 Submit the customized JCL to unload the Assure product files from the installation cartridge.

3. Included in the JCLLIB members is a member called PDSALTER. Run this installation job first.

Before you run the job, you need to supply all the input customization parameters. For example, all the members in the JCLLIB have YOURJOB coded as the default jobname. Code the parameter as shown in the example below to replace YOURJOB with the string you specify ("DEVJAD" in the example).

```
//SYSIN DD *
YOURJOB<DEVJAD<
```

You need to edit these parameters, which the PDSALTER member codes during installation:

Parameter	Entry
YOUR.JOB	Enter a jobname.
YOUR.ACCT	Enter job accounting parameters.
NOTIFYID	Enter the id to be notified.
YOUR.MSGCLASS	Enter the JES sysout message class.
YOUR.CLASS	Enter the JES job submission class.
YOUR.ASSURE.LOADLIB	Enter the Assure loadlib from installation
YOUR.BASELINE.FIELD.LEVEL.FILE	Enter your base field level file.
YOUR.BASELINE.RECORD.LEVEL.FILE	Enter your base record level file.
YOUR.TEST.VERSION.FIELD.LEVEL.FILE	Enter your test field level file.
YOUR.TEST.VERSION.RECORD.LEVEL.FILE	Enter your test record level file.
YOUR.TEST.FIELD.LEVEL.SUMMARY	Enter your test field level summary.
YOUR.TEST.RECORD.LEVEL.SUMMARY	Enter your test record level summary.
YOUR.BASELINE.FIELD.LEVEL.SUMMARY	Enter your base field level summary.
YOUR.BASELINE.RECORD.LEVEL.SUMMARY	Enter your base record level summary.
YOUR.SORT.PROGRAM	Enter your sort program.
YOUR.COMPARE.PROGRAM	Enter your compare program.
YOUR.DASD	Enter your DASD device id.

3

Creating the Assure How-changed Files

This chapter discusses the steps necessary to create the Assure how-changed files in ASG-TRACER jobstreams.

Preparing ASG-TRACER

To create the //TCRSUMRY, //SUPCRECL, and //SUPCFLDL files using the Assure user exits, you must make several changes to the standard ASG-TRACER jobs.

Editing the @TE0BREF and @TG0BRRF JCL Jobstreams

You will need to change one of these JCL jobstreams:

- If you want to include update information created by aborted transactions, change the jobstream that reads the journal archive tape—@TE0BREF.
- If you want to include update information created by only recovery units that ended successfully, change the jobstream that reads the earlier-produced ASG-TRACER reformatted file—@TG0BRRF.

Insert the information in Figure 3 on page 11 in the CLEANUP step (where the existing datasets are erased).

```
/*-BEGIN ----- REGRESSION-TESTING-COMPARE -----
/* FOR REGRESSION-TESTING COMPARE FEATURE OF TRACER:
/*
//SUPCRECL DD DSN=TCR.Rxxx.BASELINE.CV#121.SUPCRECL,
//          SPACE=(TRK,1),DISP=(MOD,DELETE),UNIT=SYSDA
/*
//SUPCFLDL DD DSN=TCR.Rxxx.BASELINE.CV#121.SUPCFLDL,
//          SPACE=(TRK,1),DISP=(MOD,DELETE),UNIT=SYSDA
/*
/*-END ----- REGRESSION-TESTING-COMPARE -----
```

Figure 3

Insert the information in Figure 4 on page 12 in these steps:

- Step RUNRJRN (in job @TE0BREF).
- Step RUNTRKR (in job @TG0BRRF).

```

/*-BEGIN ----- REGRESSION-TESTING-COMPARE -----
/* FOR REGRESSION-TESTING COMPARE FEATURE OF TRACER:
//SUPCRECL DD DSN=TCR.Rxxx.BASELINE.CV#121.SUPCRECL,
/**      DCB=(RECFM=FB,LRECL=300,BLKSIZE=?),      (3380)
/**      DCB=(RECFM=FB,LRECL=300,BLKSIZE=?),      (3390)
//      DCB=(RECFM=FB,LRECL=300,BLKSIZE=22800),      (9345)
//      SPACE=(TRK,(1,1),RLSE),
//      DISP=(,CATLG,CATLG),
/**      VOL=SER=DBS001,
//      UNIT=SYSDA
//
//SUPCFLDL DD DSN=TCR.Rxxx.BASELINE.CV#121.SUPCFLDL,
/**      DCB=(RECFM=FB,LRECL=250,BLKSIZE=?),      (3380)
/**      DCB=(RECFM=FB,LRECL=250,BLKSIZE=?),      (3390)
//      DCB=(RECFM=FB,LRECL=200,BLKSIZE=22800),      (9345)
//      SPACE=(TRK,(5,5),RLSE),
//      DISP=(,CATLG,CATLG),
/**      VOL=SER=DBS001,
//      UNIT=SYSDA
/*-END ----- REGRESSION-TESTING-COMPARE -----

```

Figure 4

Editing the //DBTINPUT Parameter File

Insert these parameters in the //DBTINPUT parameter file:

TRKR-YN-CALL-TCRRECLV	Y Y=PRODUCES //SUPCRECL FILE
TRKR-YN-CALL-TCRFLDLV	Y Y=PRODUCES //SUPCFLDL FILE
YN-CALL-TCRSUMRY	Y Y=PRODUCED //TCRSUMRY FILE
YN-USERID-IN-TCRSUMRY	N Y=BREAKS DOWN BY USERID
TRKR-RPT-YN-80-COL-MODE	N N=BEFORE AND AFTER ON SAME LINE

Result Files

Once you run the modified ASG-TRACER jobs, you will have three files. You use these files as input to your compare facility (e.g., IBM's SUPERC):

DD Name	Suggested DSN Name	Description
//TCRSUMRY	TCR.Rxx.CV#xxx.scenario.TCRSUMRY	Counts records updated by USERID, PROGRAM, Verb-type
//SUPCRECL	TCR.ASR.Rxx.CV#xxx.scenario.SUPCRECL	One row per record occurrence updated (record level)
//SUPCFLDL	TCR.ASR.Rxx.CV#xxx.scenario.SUPCFLDL	One row per record element updated (field level)

where:

CV#xxx is your Central Version number.

scenario is the naming convention of the test scenario you develop.

Once you have the Assure files from your baseline run, you can make modifications to the programs in your information system (e.g., Y2K changes, new software, etc.). After you complete your changes, you need to create the //TCRSUMRY, //SUPCRECL, and //SUPCFLDL files for the test version run.

You now have two sets of Assure files—one from your baseline run and one from the test version run. You can use the Assure summary and comparison programs to determine the differences between your files before you run your comparison utility, or you can immediately input these files into your comparison utility. In either case, the utility determines if and how your information system was updated differently when the new test versions were executed.

4

Using the Assure Summary Reports

Assure provides summary reports that allow you to perform fast comparisons between two runs of data. This comparison is performed through the use of Assure programs when combined with the provided JCL libraries.

After you produce the Assure files, you can proceed with the assurance process. You perform this process by running these programs:

- TCRPSUMM
- TCRPCOMP
- TCRFSUMM
- TCRFCOMP

TCRPSUMM—Processing Summary Report

This program takes the //SUPCRECL format file and summarizes the data present in the file. The summary is produced as a report and also as an intermediate record level summary file. This file is in a simple comma-separated form that can be used as input to the TCRPCOMP program or downloaded and used in a spreadsheet product.

Figure 5 on page 16 is an example of a simple JCL jobstream that can be used to run the program. You can find a sample of the TCRPSUMM JCL in the RUNPSUMM member of the Assure JCLLIB.

```
//YOUR.JOB JOB (YOUR.ACCT),'RUN PROCESS SUMM',
//          MSGLEVEL=(1,1),MSGCLASS=YOUR.MSGCLASS,
//          CLASS=YOUR.CLASS,NOTIFY=NOTIFYID
// *
//TCRPSUMM EXEC PGM=TCRPSUMM,REGION=0M
// *
//STEPLIB   DD DSN=YOUR.ASSURE.LOADLIB,DISP=SHR
// *
//SUPCRECL  DD DSN=YOUR.TEST.VERSION.RECORD.LEVEL.FILE,DISP=SHR
// *
//SUMMREPT  DD SYSOUT=*
// *
//SUMMCMPR  DD DSN=YOUR.TEST.RECORD.LEVEL.SUMMARY,DISP=(,CATLG,KEEP),
//          UNIT=YOUR.DASD,
//          DCB=(RECFM=FB,BLKSIZE=19500,LRECL=39),SPACE=(CYL,(1))
```

```

// *
//SYSLST DD SYSOUT=*
// *
//SYSUDUMP DD SYSOUT=*
// *
//SYSOUT DD SYSOUT=*

```

Figure 5

The program also produces a very simple summary file that has these values:

Value	Description
NB	All fields are comma-separated for easy transport to a PC and spreadsheet format.
TYPE	Summary record type. The possible values are P: for program, R: for record, V: for verb, W: for sectional breakdown, and X: for expanded breakdown. The first four types are mainly information and give a brief overview of the types of processing that have occurred. The expanded breakdown gives actual totals based on the primary processing elements of Program, Record, and Verb.
PROGRAM	Program name.
RECORD	Record name.
VERB	Verb action performed.
COUNT	Counter of occurrence of each type.

TCRPCOMP—Comparison Summary Report

This program takes the sorted SUMMCMPR files from the two runs of the TCRPSUMM program and compares the results. Based on the values in the summary file and the selected action from the parameter card file, the program produces a report of potential problem areas. In addition, it produces sort parameters to OMIT or INCLUDE records for the comparison utility selection.

A sample of the JCL to run the program is shown in Figure 6 on page 17. You can find a sample of the TCRPCOMP JCL in the RUNPCOMP member of the Assure JCLLIB.

```

//YOUR.JOB JOB (YOUR.ACCT),'RUN PROCESS COMPARE',
//          MSGLEVEL=(1,1),MSGCLASS=YOUR.MSGCLASS,
//          CLASS=YOUR.CLASS,NOTIFY=NOTIFYID
// *
// *
//TCRPCOMP EXEC PGM=TCRPCOMP,REGION=8M
// *
//STEPLIB DD DISP=SHR,DSN=YOUR.ASSURE.LOADLIB
// *
//SUMMCPA DD DISP=SHR,DSN=YOUR.BASE.RECORD.LEVEL.SUMMARY

```

```
// *
//SUMMCMPB DD DISP=SHR,DSN=YOUR.TEST.RECORD.LEVEL.SUMMARY
// *
//SORTCARD DD DISP=SHR,DSN=YOUR.SORTCARD,DISP=(,PASS),UNIT=YOUR.DASD,
//          DCB=(RECFM=FB,BLKSIZE=8000,LRECL=80),
//          SPACE=(TRK,(1))
// *
//COMPREP DD SYSOUT=*
// *
//SYSLST DD SYSOUT=*
// *
//SYSUDUMP DD SYSOUT=*
// *
//SYSOUT DD SYSOUT=*
```

Figure 6

The COMPREP contains a listing of exception conditions only and will not be produced if there are no problems or mismatches encountered in the comparison.

The SORTCARD file contains the necessary parameters to either exclude or include the potentially mismatched records from the comparison utility files before passing them to the comparison utility.

Figure 7 on page 17 is an example of the sortcards that will be produced.

```
SORT FIELDS=(COPY)
OMIT COND=(( (1,8,CH,EQ,C'CCD0117 '),AND,(10,2,CH,EQ,C'CO '),AND,
            (13,16,CH,EQ,C'CHARGE ')),
OR,
            ((1,8,CH,EQ,C'CCD0117 '),AND,(10,2,CH,EQ,C'ST '),AND,
            (13,16,CH,EQ,C'CHARGE ')))
END SORT
```

Figure 7

Combining the Process Reports

You can combine the standard IBM utilities and the Assure programs to minimize the output for the comparison report. This is all tied together through the use of a complete JCL jobstream, as shown in Figure 8 on page 21. You can find a sample of the combined JCL in the FULLRUNP member of the Assure JCLLIB.

```
//YOUR.JOB JOB (YOUR.ACCT),'RUN FULL PROCESS',
//          MSGLEVEL=(1,1),MSGCLASS=YOUR.MSGCLASS,
//          CLASS=YOUR.CLASS,NOTIFY=NOTIFYID
//*****
//* STEP 1 - RUN BASELINE SUPERF FILE INTO SUMMARY FILE
//*****
//SUMMA EXEC PGM=TCRPSUMM,REGION=0M
// *
//STEPLIB DD DSN=YOUR.ASSURE.LOADLIB,DISP=SHR
```

```

/*
//SUPCRECL DD DSN=YOUR.BASELINE.RECORD.LEVEL.FILE,DISP=SHR
/*
//SUMMREPT DD SYSOUT=*
/*
//SUMMCMPR DD DSN=&&COMPA,DISP=( ,PASS,KEEP) ,
//          UNIT=YOUR.DASD,
//          DCB=(RECFM=FB,BLKSIZE=19500,LRECL=39) ,SPACE=(CYL,(1))
/*
//SYSLST DD SYSOUT=*
/*
//SYSUDUMP DD SYSOUT=*
/*
//SYSOUT DD SYSOUT=*
//*****
/* STEP 1A - SORT INTERMEDIATE SUMMARY FILE A
//*****
//SORTA EXEC PGM=YOUR.SORT.PROGRAM,REGION=0M
/*
//SYSOUT DD SYSOUT=*
/*
//SORTIN DD DSN=&&COMPA,DISP=(OLD,DELETE)
/*
//SORTOUT DD DSN=&&SORTA,DISP=( ,PASS,KEEP) ,
//          UNIT=YOUR.DASD,SPACE=(CYL,(1)) ,
//          DCB=(RECFM=FB,BLKSIZE=19500,LRECL=39)
/*
//SORTWK01 DD DSN=&&WK01,UNIT=YOUR.DASD,DISP=( ,DELETE) ,SPACE=(CYL,(2,2))
/*
//SORTWK02 DD DSN=&&WK02,UNIT=YOUR.DASD,DISP=( ,DELETE) ,SPACE=(CYL,(2,2))
/*
//SORTWK03 DD DSN=&&WK03,UNIT=YOUR.DASD,DISP=( ,DELETE) ,SPACE=(CYL,(2,2))
/*
//SORTSNAP DD SYSOUT=*
/*
//SYSUDUMP DD SYSOUT=*
/*
//SYSMDUMP DD SYSOUT=*
/*
//SYSABEND DD SYSOUT=*
/*
//SYSIN DD *
SORT FIELDS=(1,39,CH,A)
END SORT
/*
//*****
/* STEP 2 - RUN TESTVERS SUPERC FILE INTO SUMMARY FILE
//*****
//SUMMB EXEC PGM=TCRPSUMM,REGION=0M
/*
//STEPLIB DD DSN=YOUR.ASSURE.LOADLIB,DISP=SHR
/*
//SUPCRECL DD DSN=YOUR.TEST.VERSION.RECORD.LEVEL.FILE,DISP=SHR
/*
//SUMMREPT DD SYSOUT=*
/*
//SUMMCMPR DD DSN=&&COMPB,DISP=( ,PASS,KEEP) ,
//          UNIT=YOUR.DASD,
//          DCB=(RECFM=FB,BLKSIZE=19500,LRECL=39) ,SPACE=(CYL,(1))
/*
//SYSLST DD SYSOUT=*
/*

```

```
//SYSUDUMP DD SYSOUT=*
//*
//SYSOUT DD SYSOUT=*
//*
//*****
//* STEP 2A - SORT INTERMEDIATE SUMMARY FILE B
//*****
//SORTB EXEC PGM=YOUR.SORT.PROGRAM,REGION=0M
//*
//SYSOUT DD SYSOUT=*
//*
//SORTIN DD DSN=&&COMPB,DISP=(OLD,DELETE)
//*
//SORTOUT DD DSN=&&SORTB,DISP=(,PASS,KEEP),
// UNIT=YOUR.DASD,SPACE=(CYL,(1)),
// DCB=(RECFM=FB,BLKSIZE=19500,LRECL=39)
//*
//SORTWK01 DD DSN=&&WK01,UNIT=YOUR.DASD,DISP=(,DELETE),SPACE=(CYL,(2,2))
//*
//SORTWK02 DD DSN=&&WK02,UNIT=YOUR.DASD,DISP=(,DELETE),SPACE=(CYL,(2,2))
//*
//SORTWK03 DD DSN=&&WK03,UNIT=YOUR.DASD,DISP=(,DELETE),SPACE=(CYL,(2,2))
//*
//SORTSNAP DD SYSOUT=*
//*
//SYSUDUMP DD SYSOUT=*
//*
//SYSMDUMP DD SYSOUT=*
//*
//SYSABEND DD SYSOUT=*
//*
//SYSIN DD *
SORT FIELDS=(1,39,CH,A)
END SORT
//*
//*****
//* STEP 3 - RUN PROCESSING COMPARISON
//*****
//*
//TCRPCOMP EXEC PGM=TCRPCOMP,REGION=8M
//*
//STEPLIB DD DISP=SHR,DSN=YOUR.ASSURE.LOADLIB
//*
//PARMFILE DD *
OMIT/INCLUDE
//*
//SUMMCPA DD DSN=&&SORTA,DISP=(OLD,DELETE)
//*
//SUMMCMPB DD DSN=&&SORTB,DISP=(OLD,DELETE)
//*
//SORTCARD DD DSN=&&SORTCRDA,DISP=(,PASS),UNIT=YOUR.DASD,
// DCB=(RECFM=FB,BLKSIZE=8000,LRECL=80),
// SPACE=(TRK,(1))
//*
//COMPREP DD SYSOUT=*
//*
//SYSLST DD SYSOUT=*
//*
//SYSUDUMP DD SYSOUT=*
//*
//SYSOUT DD SYSOUT=*
//*
```

```
//*****  
//* STEP 4 - INCLUDE/EXCLUDE FOR BASELINE SUPCRECL FILE  
//*****  
//*  
//SORTXA EXEC PGM=YOUR.SORT.PROGRAM,REGION=0M  
//*  
//SYSOUT DD SYSOUT=*  
//*  
//SORTIN DD DSN=YOUR.BASELINE.RECORD.LEVEL.FILE,DISP=SHR  
//*  
//SORTOUT DD DSN=&&EXTRACTA,DISP=(,PASS,KEEP),  
// UNIT=YOUR.DASD,SPACE=(CYL,(1)),  
// DCB=(RECFM=FB,BLKSIZE=19500,LRECL=39)  
//*  
//SORTWK01 DD DSN=&&WK01,UNIT=YOUR.DASD,DISP=(,DELETE),SPACE=(CYL,(2,2))  
//*  
//SORTWK02 DD DSN=&&WK02,UNIT=YOUR.DASD,DISP=(,DELETE),SPACE=(CYL,(2,2))  
//*  
//SORTWK03 DD DSN=&&WK03,UNIT=YOUR.DASD,DISP=(,DELETE),SPACE=(CYL,(2,2))  
//*  
//SORTSNAP DD SYSOUT=*  
//*  
//SYSUDUMP DD SYSOUT=*  
//*  
//SYSMDUMP DD SYSOUT=*  
//*  
//SYSABEND DD SYSOUT=*  
//*  
//SYSIN DD DSN=&&SORTCRDA,DISP=(OLD,PASS)  
//*****  
//* STEP 5 - INCLUDE/EXCLUDE FOR TEST VERSION SUPCRECL FILE  
//*****  
//*  
//SORTXB EXEC PGM=YOUR.SORT.PROGRAM,REGION=0M  
//*  
//SYSOUT DD SYSOUT=*  
//*  
//SORTIN DD DSN=YOUR.TEST.VERSION.RECORD.LEVEL.FILE,DISP=SHR  
//*  
//SORTOUT DD DSN=&&EXTRACTB,DISP=(,PASS,KEEP),  
// UNIT=YOUR.DASD,SPACE=(CYL,(1)),  
// DCB=(RECFM=FB,BLKSIZE=19500,LRECL=39)  
//*  
//SORTWK01 DD DSN=&&WK01,UNIT=YOUR.DASD,DISP=(,DELETE),SPACE=(CYL,(2,2))  
//*  
//SORTWK02 DD DSN=&&WK02,UNIT=YOUR.DASD,DISP=(,DELETE),SPACE=(CYL,(2,2))  
//*  
//SORTWK03 DD DSN=&&WK03,UNIT=YOUR.DASD,DISP=(,DELETE),SPACE=(CYL,(2,2))  
//*  
//SORTSNAP DD SYSOUT=*  
//*  
//SYSUDUMP DD SYSOUT=*  
//*  
//SYSMDUMP DD SYSOUT=*  
//*  
//SYSABEND DD SYSOUT=*  
//*  
//SYSIN DD DSN=&&SORTCRDA,DISP=(OLD,DELETE)  
//*  
//*****  
//* STEP 6 - SUPERCOMPARISON  
//*****
```



```
// *
//CMPRECL EXEC PGM=YOUR.COMPARE.PROGRAM,
//          PARM=(DELTAL,LINECMP,LOGLN,NOPRTCC)
// *
//NEWDD DD DSN=&&EXTRACTB,DISP=(OLD,DELETE)
// *
//OLDDDD DD DSN=&&EXTRACTA,DISP=(OLD,DELETE)
// *
//OUTDD DD SYSOUT=*
// *
//SYSIN DD *
CMPCOLM 1:78
```

Figure 8

TCRFSUMM—Element Summary Report

Running against the //SUPCFLVL file, this program produces a report of the records, elements, and verbs that have been used by the processing being captured. The program also produces an interim file that contains a breakdown of the processing at each of the levels described. This file is then used as input to the TCRFCOMP process.

A sample of the JCL to run the program is shown in Figure 9 on page 21. You can find a sample of the TCRFSUMM JCL in the RUNFCOMP member of the Assure JCLLIB.

```
//YOUR.JOB JOB (YOUR.ACCT),'RUN FIELD SUMMARY',
//          MSGLEVEL=(1,1),MSGCLASS=YOUR.MSGCLASS,
//          CLASS=YOUR.CLASS,NOTIFY=NOTIFYID
// *
//TCRFSUMM EXEC PGM=TCRFSUMM,REGION=0M
// *
//STEPLIB DD DSN=YOUR.ASSURE.LOADLIB,DISP=SHR
// *
//SUPCFLDL DD DSN=YOUR.TEST.VERSION.FIELD.LEVEL.FILE,DISP=SHR
// *
//SUMMREPT DD SYSOUT=*
// *
//SUMMCMPF DD DSN=YOUR.TEST.FIELD.LEVEL.SUMMARY,DISP=(,CATLG,KEEP),
//          DCB=(RECFM=FB,BLKSIZE=6300,LRECL=63),SPACE=(CYL,(1)),
//          UNIT=YOUR.DASD
// *
//SYSLST DD SYSOUT=*
// *
//SYSUDUMP DD SYSOUT=*
// *
//SYSOUT DD SYSOUT=*
```

Figure 9

TCRFCOMP—Element Comparison Summary

As with the TCRPCOMP, this program compares the interim field level summary files and, depending on parameter input, allows input to the SORT process.

A sample of the JCL to run the program is shown in Figure 10 on page 22. You can find a sample of the TCRFCOMP JCL in the RUNFSUMM member of the Assure JCLLIB.

```
//YOUR.JOB JOB (YOUR.ACCT),'RUN ELEMENT COMPARE',
//          MSGLEVEL=(1,1),MSGCLASS=YOUR.MSGCLASS,
//          CLASS=YOUR.CLASS,NOTIFY=NOTIFYID
// *
//TCRFCOMP EXEC PGM=TCRFCOMP,REGION=8M
// *
//STEPLIB DD DISP=SHR,DSN=YOUR.ASSURE.LOADLIB
// *
//PARMFILE DD *
OMIT/INCLUDE
// *
//SUMMCPA DD DISP=SHR,DSN=YOUR.BASE.FIELD.LEVEL.SUMMARY
// *
//SUMMCPB DD DISP=SHR,DSN=YOUR.TEST.FIELD.LEVEL.SUMMARY
// *
//SORTCARD DD DISP=SHR,DSN=&&SORTCRDA,DISP=(,PASS),UNIT=YOUR.DASD,
//          DCB=(RECFM=FB,BLKSIZE=8000,LRECL=80),
//          SPACE=(TRK,(1))
// *
//COMPREPT DD SYSOUT=*
// *
//SYSLST DD SYSOUT=*
// *
//SYSUDUMP DD SYSOUT=*
// *
//SYSOUT DD SYSOUT=*
// *
```

Figure 10

Combining the Field Reports

You can combine the standard IBM utilities and the Assure programs to minimize the comparison report output. This is all tied together through the use of a complete JCL jobstream, as shown in Figure 11 on page 26. You can find a sample of the combined JCL in the FULLRUNF member of the Assure JCLLIB.

```
//YOUR.JOB JOB (YOUR.ACCT),'RUN FULL FIELD',
//          MSGLEVEL=(1,1),MSGCLASS=YOUR.MSGCLASS,
//          CLASS=YOUR.CLASS,NOTIFY=NOTIFYID
//*****
// * STEP 1 - RUN BASELINE SUPCFLDL INTO SUMMARY FILE
//*****
// *
//SUMMA EXEC PGM=TCRFSUMM,REGION=0M
```

```

// *
// STEPLIB DD DSN=YOUR.ASSURE.LOADLIB,DISP=SHR
// *
// SUPCFLDL DD DSN=YOUR.BASELINE.FIELD.LEVEL.FILE,DISP=SHR
// *
// SUMMREPT DD SYSOUT=*
// *
// SUMMCMPF DD DSN=&&COMPA,DISP=(,PASS,KEEP),
// UNIT=YOUR.DASD,
// DCB=(RECFM=FB,BLKSIZE=12600,LRECL=63),SPACE=(CYL,(1))
// *
// SYSLST DD SYSOUT=*
// *
// SYSUDUMP DD SYSOUT=*
// *
// SYSOUT DD SYSOUT=*
// *
// *****
// * STEP 1A - SORT INTERMEDIATE SUMMARY FILE A
// *****
// *
// SORTA EXEC PGM=YOUR.SORT.PROGRAM,REGION=0M
// *
// SYSOUT DD SYSOUT=*
// *
// SORTIN DD DSN=&&COMPA,DISP=(OLD,DELETE)
// *
// SORTOUT DD DSN=&&SORTA,DISP=(,PASS,KEEP),
// UNIT=YOUR.DASD,SPACE=(CYL,(1)),
// DCB=(RECFM=FB,BLKSIZE=12600,LRECL=63)
// *
// SORTWK01 DD DSN=&&WK01,UNIT=YOUR.DASD,DISP=(,DELETE),SPACE=(CYL,(2,2))
// *
// SORTWK02 DD DSN=&&WK02,UNIT=YOUR.DASD,DISP=(,DELETE),SPACE=(CYL,(2,2))
// *
// SORTWK03 DD DSN=&&WK03,UNIT=YOUR.DASD,DISP=(,DELETE),SPACE=(CYL,(2,2))
// *
// SORTSNAP DD SYSOUT=*
// *
// SYSUDUMP DD SYSOUT=*
// *
// SYSMDUMP DD SYSOUT=*
// *
// SYSABEND DD SYSOUT=*
// *
// SYSIN DD *
// SORT FIELDS=(1,63,CH,A)
// END SORT
// *
// *****
// * STEP 2 - RUN SECOND SUPERF FILE INTO SUMMARY FILE
// *****
// *
// SUMMB EXEC PGM=TCRFSUMM,REGION=0M
// *
// STEPLIB DD DSN=YOUR.ASSURE.LOADLIB,DISP=SHR
// *
// SUPCFLDL DD DSN=YOUR.TEST.VERSION.FIELD.LEVEL.FILE,DISP=SHR
// *
// SUMMREPT DD SYSOUT=*
// *
// SUMMCMPF DD DSN=&&COMPB,DISP=(,PASS,KEEP),

```

```
//          UNIT=YOUR.DASD,
//          DCB=(RECFM=FB, BLKSIZE=12600, LRECL=63), SPACE=(CYL,(1))
// *
//SYSLST    DD SYSOUT=*
// *
//SYSUDUMP  DD SYSOUT=*
// *
//SYSOUT    DD SYSOUT=*
// *
//*****
// * STEP 2A - SORT INTERMEDIATE SUMMARY FILE B
//*****
// *
//SORTB     EXEC PGM=YOUR.SORT.PROGRAM, REGION=0M
// *
//SYSOUT    DD SYSOUT=*
// *
//SORTIN    DD DSN=&&COMPB, DISP=(OLD,DELETE)
// *
//SORTOUT    DD DSN=&&SORTB, DISP=(,PASS,KEEP),
//            UNIT=YOUR.DASD, SPACE=(CYL,(1)),
//            DCB=(RECFM=FB, BLKSIZE=12600, LRECL=63)
// *
//SORTWK01   DD DSN=&&WK01, UNIT=YOUR.DASD, DISP=(,DELETE), SPACE=(CYL,(2,2))
// *
//SORTWK02   DD DSN=&&WK02, UNIT=YOUR.DASD, DISP=(,DELETE), SPACE=(CYL,(2,2))
// *
//SORTWK03   DD DSN=&&WK03, UNIT=YOUR.DASD, DISP=(,DELETE), SPACE=(CYL,(2,2))
// *
//SORTSNAP   DD SYSOUT=*
// *
//SYSUDUMP   DD SYSOUT=*
// *
//SYSMDUMP   DD SYSOUT=*
// *
//SYSABEND   DD SYSOUT=*
// *
//SYSIN      DD *
//            SORT FIELDS=(1,63,CH,A)
//            END SORT
// *
//*****
// * STEP 3 - RUN PROCESSING COMPARISON
//*****
// *
//TCRFCOMP   EXEC PGM=TCRFCOMP, REGION=8M
// *
//STEPLIB    DD DISP=SHR, DSN=YOUR.ASSURE.LOADLIB
// *
//PARMFILE   DD *
//            OMIT/INCLUDE
// *
//SUMMCMPA   DD DSN=&&SORTA, DISP=(OLD,DELETE)
// *
//SUMMCMPB   DD DSN=&&SORTB, DISP=(OLD,DELETE)
// *
//SORTCARD   DD DSN=&&SORTCRDA, DISP=(,PASS), UNIT=YOUR.DASD,
//            DCB=(RECFM=FB, BLKSIZE=8000, LRECL=80),
//            SPACE=(TRK,(1))
// *
//COMPREPT   DD SYSOUT=*
// *
```

```
//SYSLST      DD SYSOUT=*
//*
//SYSUDUMP    DD SYSOUT=*
//*
//SYSOUT      DD SYSOUT=*
//*
//*****
/* STEP 4 - INCLUDE/EXCLUDE FOR FILE A
//*****
//*
//SORTXA      EXEC PGM=YOUR.SORT.PROGRAM,REGION=0M
//*
//SYSOUT      DD SYSOUT=*
//*
//SORTIN      DD DSN=YOUR.BASELINE.FIELD.LEVEL.FILE,DISP=SHR
//*
//SORTOUT     DD DSN=&&EXTRACTA,DISP=( ,PASS,KEEP) ,
//              UNIT=YOUR.DASD,SPACE=(CYL,(1)),
//              DCB=(RECFM=FB,BLKSIZE=12600,LRECL=63)
//*
//SORTWK01    DD DSN=&&WK01,UNIT=YOUR.DASD,DISP=( ,DELETE),SPACE=(CYL,(2,2))
//*
//SORTWK02    DD DSN=&&WK02,UNIT=YOUR.DASD,DISP=( ,DELETE),SPACE=(CYL,(2,2))
//*
//SORTWK03    DD DSN=&&WK03,UNIT=YOUR.DASD,DISP=( ,DELETE),SPACE=(CYL,(2,2))
//*
//SORTSNAP    DD SYSOUT=*
//*
//SYSUDUMP    DD SYSOUT=*
//*
//SYSMDUMP    DD SYSOUT=*
//*
//SYSABEND    DD SYSOUT=*
//*
//SYSIN       DD DSN=&&SORTCRDA,DISP=(OLD,PASS)
//*
//*****
/* STEP 5 - INCLUDE/EXCLUDE FOR FILE B
//*****
//*
//SORTXB      EXEC PGM=YOUR.SORT.PROGRAM,REGION=0M
//*
//SYSOUT      DD SYSOUT=*
//*
//SORTIN      DD DSN=YOUR.TEST.VERSION.FIELD.LEVEL.FILE,DISP=SHR
//*
//SORTOUT     DD DSN=&&EXTRACTB,DISP=( ,PASS,KEEP) ,
//              UNIT=YOUR.DASD,SPACE=(CYL,(1)),
//              DCB=(RECFM=FB,BLKSIZE=12600,LRECL=63)
//*
//SORTWK01    DD DSN=&&WK01,UNIT=YOUR.DASD,DISP=( ,DELETE),SPACE=(CYL,(2,2))
//*
//SORTWK02    DD DSN=&&WK02,UNIT=YOUR.DASD,DISP=( ,DELETE),SPACE=(CYL,(2,2))
//*
//SORTWK03    DD DSN=&&WK03,UNIT=YOUR.DASD,DISP=( ,DELETE),SPACE=(CYL,(2,2))
//*
//SORTSNAP    DD SYSOUT=*
//*
//SYSUDUMP    DD SYSOUT=*
//*
//SYSMDUMP    DD SYSOUT=*
//*
```

```
//SYSABEND DD SYSOUT=*
//*
//SYSIN DD DSN=&&SORTCRDA,DISP=(OLD,DELETE)
//*
//*****
//* STEP 6 - SUPERC COMPARISON
//*****
//*
//CMPRECL EXEC PGM=YOUR.COMPARE.PROGRAM,
//          PARM=(DELTAL,LINECMP,LOGLN,NOPRTCC)
//*
//NEWDD DD DSN=&&EXTRACTB,DISP=(OLD,DELETE)
//*
//OLDDDD DD DSN=&&EXTRACTA,DISP=(OLD,DELETE)
//*
//OUTDD DD SYSOUT=*
//SYSIN DD *
CMPCOLM 1:116
```

Figure 11

Appendix A

//TCRSUMRY File

The ASG-TRACER summary file shows the number of times that each processed record type is updated. A typical record in the summary file will contain six literals, each followed by a variable.

U:USERID

Indicates the USERID that was active for either the online task or CV-batch job under which the PROGRAM issued the update VERB. The USERID begins in column 5 of the //TCRSUMRY file, immediately following the U: literal.

R:RECORD-NAME

Indicates the name of the RECORD that was updated. RECORD-NAME begins in column 16 of the //TCRSUMRY file, immediately following the R: literal.

P:PROGRAM-NAME

Indicates the name of the PROGRAM that issued the update VERB. PROGRAM-NAME begins in column 35 of the //TCRSUMRY file, immediately following the P: literal

V:VERB-TYPE-OF-UPDATE

Indicates a two-digit abbreviation describing the type of IDMS update VERB the updating program issued. VERB-TYPE-OF-UPDATE begins in column 46 of the //TCRSUMRY file, immediately following the V: literal.

Possible values are:

Code	Meaning
MO	MODIFY
ST	STORE
ER	ERASE (normal)
EP	ERASE PERMANENT
ES	ERASE SELECTIVE
EA	ERASE ALL

Code	Meaning
DI	DISCONNECT
CO	CONNECT
AL	All VERBs [Used only on lines where the count (#:) is for a total of all verb types used to update the record name.]

#:COUNT

Indicates the number of times the indicated VERB updated the indicated record type. COUNT begins in column 51 of the //TCRSUMRY file, immediately following the #: literal.

S:SET-NAME

Indicates the name of the SET that the program specified when it issued the SET-oriented update VERB. This field only contains a value when the type of VERB is either CONNECT or DISCONNECT. SET-NAME begins in column 63 of the //TCRSUMRY file, immediately following the S: literal.

Appendix B

//SUPCRECL File

This section is documentation for the file //SUPCRECL—the record-level file that is designed for use with your comparison utility for purposes of regression testing (for instance with Y2K final testing) and/or for downloading another computing platform (perhaps a PC) for processing in that arena.

RL-PROGRAM-NAME

Indicates the name of the program that did the access of the database record. RL-PROGRAM-NAME is located in column 1 of the //SUPCRECL file.

RL-VERB-NAME

Indicates a two-digit code to describe what type of verb was used by the application program associated with this data access. RL-VERB-NAME is located in column 10 of the //SUPCRECL file.

Possible values are:

Code	Meaning
MO	MODIFY
ST	STORE
ER	ERASE
EP	ERASE PERMANENT
ES	ERASE SELECTIVE
EA	ERASE ALL
DI	DISCONNECT
CO	CONNECT
GE	GET (not yet implemented in original 5.0.3 release of ASG-TRACER)

RL-RECORD-NAME

Indicates the name of the IDMS RECORD that was accessed. RL-RECORD-NAME is located in column 13 of the //SUPCRECL file.

RL-1ST-16-BYTES-DISPLAY

Indicates the first 16 bytes of displayable data from the record (or if only the pointers were journaled) SET-NAME. RL-1ST-16-BYTES-DISPLAY is located in column 30 of the //SUPCRECL file.

If data was journaled (i.e., the verb is either a MODIFY, or a STORE, or an ERASE), then this is the first 16 characters of the data portion of the record occurrence in as displayable a format as possible. If one of the fields in the first 16 bytes of the record is defined as a data type of COMP or COMP-3 so that the character is not printable, then a period (.) is placed in the corresponding positions of the non-printable field. You will be able to understand what is actually in these positions by looking at the next field, 1ST-16-BYTES-IN-HEXADECIMAL. This shows the value of the characters in side-by-side hexadecimal format).

If the verb is either a CONNECT or a DISCONNECT, then this will contain the name of the SET associated with this update.

Note: _____

Several of the columns contain different types of values if the update verb causes only pointers (DBKEYS) to be journaled.

RL-1ST-16-BYTES-HEXADECIMAL

Indicates the first 16 bytes of hexadecimal data from the record or message.

RL-1ST-16-BYTES-HEXADECIMAL is located in column 47 of the //SUPCRECL file.

When data is journaled (STORE, MODIFY, or ERASE), then this contains the hexadecimal representation of the contents of the first 16 bytes of the data portion of the database record. If the verb is a CONNECT or DISCONNECT and hence only pointers are journaled then this will always contain the value SEE_OWNER_DBKEY_OF_CHNGD_SET. The reason for this message is to point your attention to another field (in column 157), which will contain the OWNER DBKEY for the set that was affected by either the CONNECT or the DISCONNECT verb.

Note: _____

If the SET is not defined in the SCHEMA with OWNER pointers, then the field in column 157 will contain the NEXT DBKEY for the SET. The value in column 155 will be N (for NEXT) instead of O (for OWNER).

RL-JRNL-REC-TYPE

Indicates the type indicator from a before or after image. RL-JRNL-REC-TYPE is located in column 80 of the //SUPCRECL file.

When the verb type is MODIFY, ASG-TRACER writes two record occurrences to this //SUPCRECL file. One has the value B (for before image), and the other A (for after image). The B line shows the data in the before image of the record occurrence that was journaled and the A line shows the data in the after image. The compare when working against this file would only flag situations where there was different data changed in the same logical record occurrence if one of the fields that was changed in a different manner was physically located in the first 16 bytes of the record. (When the compare process is done using the other file //SUPCFLDL, any difference in fields throughout the record will be flagged.)

When the verb is STORE, only the A value will be in this type indicator, and when it is an ERASE the B value will be here. If the verb is CONNECT, this type indicator is set to C. It is set to D when the verb is DISCONNECT.

RL-MODIFY-1ST-DIFF-OFFSET

Indicates a changed offset, the offset of the first changed byte. The RL-MODIFY-1ST-DIFF-OFFSET is located in column 82 of the //SUPCRECL file.

When the verb is a MODIFY, this is the offset of the first byte in the record that is different after the MODIFY is done versus the way the record was before the MODIFY. ASG suggests that the comparison utility stop at column 80. However, if you do not use database procedures that put a date-time stamp at the beginning of the record, then this field could also be included in the portion of the records that your comparison utility compares. This way, other differences could be found as to how the record was updated.

RL-USERID

Indicates the USERID of the person running the program. RL-USERID is located in column 88 of the //SUPCRECL file.

This field is only filled when the ASG-TRACER exit-14 has been installed in the IDMS Central Version that produced the journal archive tape, or in the RHDCUXIT module that is used by a local-mode update job that produced the journal archive tape.

RL-RUN-UNIT

Indicates the run-unit number that issued the access verb. RL-RUN-UNIT is located in column 96 of the //SUPCRECL file.

This field contains the run-unit number of the run unit that issued the access verb.

RL-DBK-PAGE-NUMBER**RL-LINE-INDEX-NUMBER**

Indicates the DBKEY of the record that was accessed. RL-DBK-PAGE-NUMBER and RL-LINE-INDEX-NUMBER are located in column 107 of the //SUPCRECL file.

This is actually two fields, but together they constitute the DBKEY of the record that was accessed. The PAGE-NUMBER and the LINE-INDEX make up the physical DBKEY address where this accessed record is stored on the database.

RL-THIS-DBKEY-4-SR3-RECORD-CD

Indicates that the DBKEY is for an SR3 relocated record. RL-THIS-DBKEY-4-SR3-RECORD-CD is located in column 123 of the //SUPCRECL file.

This code indicates if the DBKEY for this record is associated with a RELOCATED RECORD physically on the database. Normally, ASG-TRACER might return the DBKEY of an SR3 record when the updated record occurrence was relocated by the IDMS restructure utility.

This should not be a problem when running the comparison utility because this field is not one of those that is compared. However, if an application program reads this file and then goes back to the database and issues an OBTAIN USING DBKEY type of verb (if the field is a Y), then that program might not be able to successfully retrieve the record. This is due to the DBKEY not being the DBKEY of the SR2 record, but rather, the SR3 record type that was built by the restructure utility.

If you must have the real DBKEY in all cases, and you have records on your database that have been relocated and may be updated on the database, then you may run the special jobs to find all relocated records. ASG-TRACER always puts the DBKEY of the appropriate SR2 record in this file—even though IDMS did not write it to the journal.

Note:

The jobstreams in the setup process that prepare the VSAM file used by ASG-TRACER to get the SR2 DBKEYS are named with the convention of the first two characters starting with @R.....

These are the three possible values for this field:

Value	Description
N	The DBKEY in this record is not associated with a relocated record - (this DBKEY could be used in a DML verb: OBTAIN USING DBKEY to obtain this record).
Y	The DBKEY in this record is actually the DBKEY of the SR3 record on the database, and this is a relocated record - (this DBKEY could not be used in a DML verb: OBTAIN USING DBKEY to obtain this record).
2	The DBKEY in this record is that of the SR2 record on the database for a relocated record - (the SR2 record is the same DBKEY that would be used in a DML verb: OBTAIN USING DBKEY to obtain this record).

JRNL-SEQNC

Indicates the journal sequence number in ASG-TRACER internal format. JRNL-SEQNC is located in column 125 of the //SUPCRECL file.

This is the Journal Sequence number (as much as will completely fit in a single binary fullword). In IDMS releases 12.x and 14.x, IDMS uses a binary double word field on the journal to keep the journal sequence number. ASG-TRACER internally uses only a full word, never a number that would require more than nine digits in display form. When a site's true journal sequence number grows to be larger than a number that can be displayed as a nine digit number, ASG-TRACER internally uses the truncated last nine digits of the number. That is the number you will see in this field. When the "true" journal sequence number will fit in a single binary full word (non-negative and can be displayed in nine digits or less) then the number will be identical to the number in the double word binary field on the journal.

**RL-VIA-SET-OWN-TYP-DBK
RL-VIASET-OWNER-DBKEY**

Indicates the DBKEY of the OWNER record of the SET that this record-type is stored VIA of. RL-VIA-SET-OWN-TYP-DBK and RL-VIASET-OWNER-DBKEY are located in column 136 of the //SUPCRECL file.

There are two fields. The first field contains a one-byte indicator—either O (the DBKEY is for an OWNER DBKEY), or N (the DBKEY is a NEXT type of DBKEY). The second field is the DBKEY itself. This field is most useful if the accessed record is stored VIA a set and if the set has OWNER pointers defined. For instance, if the record-type is EMPOSITION, then you might want to know which EMPLOYEE record this EMPOSITION is associated with (if the EMPOSITION record-type is stored via the EMP-EMPOSITION set and that set is defined with OWNER pointers), then TRACER extracts the OWNER pointer for the EMP-EMPOSITION set and places it in this field.

RL-DATA-OR-POINTERS-ONLY-CD

Indicates the data changed or pointer changed type of the journal record.

RL-DATA-OR-POINTERS-ONLY-CD is located in column 153 of the //SUPCRECL file.

This one character code is set to one of these values:

Character Code	Description
----------------	-------------

D	Data for this record was journaled (and pointer DBKEYS also).
P	<p>Only pointers for the record were journaled—(such as is the case when the verb is a CONNECT or a DISCONNECT—or if a change to a set control field is done on a MODIFY verb and IDMS must change the position in sets of this record and other records in the set that are physically connected.</p> <p>Usually ASG-TRACER is run with a parameter that says to process only the target record of the DML verb. This means that the only P type of records in this file would normally be for CONNECT and DISCONNECT verbs. However, you may run ASG-TRACER asking it to show all records updated even if IDMS was automatically adjusting their pointers. When this is the case, this field could be set to a P even on MODIFY, STORE, or ERASE verbs where IDMS is adjusting SET DBKEYs automatically based on the appropriate positionings because of the value of keys that are control types for various sets with automatic connection attributes.</p>

RL-CHGD-SET-OWN-TYP-DBK RL-OWNERS-DBKEY

Indicates the changed set's OWNER's DBKEY, or when data was journaled, the SEGMENT name. RL-CHGD-SET-OWN-TYP-DBK and RL-OWNERS-DBKEY are located in column 155 of the //SUPCRECL file.

When the verb is a CONNECT or a DISCONNECT (or when only pointers are journaled) these fields are designed to tell the customer the DBKEY of the record that is the OWNER of the set that the updated record-occurrence participates in (either was CONNECTed to or DISCONNECTed from). The first byte is for an indicator that will either be O indicating the DBKEY is that of the OWNER of the set, or N indicating the DBKEY is that for the NEXT in the set when new OWNER pointers were defined in the SCHEMA for the set.

When the line is for a MODIFY, STORE, or ERASE verb and data was journaled, then the SEGMENT name that the record is in is placed in the field that would contain the DBKEY if only pointers were journaled.

RL-YN-TARGET-OF-DML-V-REC

Indicates the target of the DML verb indicator. RL-YN-TARGET-OF-DML-V-REC is located in column 172 of the //SUPCRECL file.

The indicator is set to Y or N, where these conditions apply:

Indicator	Description
Y	This record is the target of the DML verb. This means that this is the record intended when the application programmer coded the DML update verb.
N	This record was automatically updated by IDMS. This is a record-occurrence that is logically connected or disconnected in the set by IDMS automatically based upon automatic set orders. The set control key values are in the data portion of the target record (as in sets that are automatically kept in sequence based on sorted sets on specific fields in the record occurrence).

This field would only be set to N when ASG-TRACER processed the journal and this parameter is set in the //DBTINPUT file:

```
SEL-YN-ONLY-TARGET-OF-DML-RC      N Y=ONLY RECORD THAT PROG SAID UPDATE
```

The default and the setting that would normally be used in regression testing situations is Y.

RL-TIME-OF-UPDATE-HHMMSSMMMMMM

Indicates the time of the update or access, in hours, minutes, seconds, and millionths of a second. RL-TIME-OF-UPDATE-HHMMSSMMMMMM is located in column 174 of the //SUPCRECL file.

RL-TASK-CODE

Indicates the name of the TASK through which the record was accessed, or if the access was done by a batch job, the JOBNAME. RL-TASK-CODE is located in column 187 of the //SUPCRECL file.

This is only filled if the ASG-TRACER exit-14 is installed. If the program that accessed the record was running online, this contains the TASK CODE. If the program was running in batch mode this will contain the JOB-NAME for the JOB.

RL-PAGE-GROUP

Indicates the page-group that was active in the IDMS RUN-UNIT when this record was accessed. RL-PAGE-GROUP is located in column 196 of the //SUPCRECL file.

RL-DML-VERB-NBR

Indicates the count of the number of update verbs into this run unit. RL-DML-VERB-NBR is located in column 201 of the //SUPCRECL file.

This is a number that is derived as follows: ASG-TRACER increments a counter each time it encounters records on the journal that appear to be from another update DML verb for this run unit. When the TRACER parameter:

```
SEL-YN-ONLY-TARGET-OF-DML-RC      N Y=ONLY RECORD THAT PROG SAID UPDATE
```

is used, then there may be several records updated by IDMS when the application program has issued only one update DML verb. For instance when a STORE of a record is requested and that record participates in a sorted set that has the "automatic connect" option set in the SCHEMA, then IDMS must adjust the pointers of the appropriate adjacent record occurrences in the set. All of the record occurrences that IDMS updated in response to the STORE verb will have the same number in this field.

RL-DATE-OF-UPDATE-CCYYMMDD

Indicates the date of the access. RL-DATE-OF-UPDATE-CCYYMMDD is located in column 207 of the //SUPCRECL file. The date is expressed as:

- Two columns for the Century (e.g., 19 or 20)
- Two columns for the Year (e.g., 98)
- Two columns for the Month (e.g., 12)
- Two columns for the Day (e.g., 31)

RL-TERMINAL

Indicates the terminal name from where the access was initiated. RL-TERMINAL is located in column 216 of the //SUPCRECL file.

When the ASG-TRACER exit-14 is installed in the IDMS Central Version that produced the journal archive file(s), then this contains the Terminal name.

RL-TYPE-CHANGED-DBKEY

Indicates a changed pointer, the pointer which was changed in this SET.

RL-TYPE-CHANGED-DBKEY is located in column 225 of the //SUPCRECL file.

This is a one-character code that shows which pointer in the set was changed when only pointers were journaled. It will be represented by one of these character codes:

Character Code	Description
N	Next pointer
P	Prior pointer
O	Owner pointer

NEW-CHANGED-DBK

OR-DA-17-31-DIS

Indicates the new image of the DBKEY that was changed, or when data is journaled, bytes 17 through 31 of the record. The NEW-CHANGED-DBK and OR-DA-17-31-DIS are located in column 227 of the //SUPCRECL file.

When only pointers are journaled, this contains the DBKEY of the changed pointer. When data is journaled, this contains the columns 17 through 31 in as displayable a format as possible.

Note:

When there are unprintable characters, a period is present in the appropriate position.

RL-ALT-IMAGE-CHNGD-DBKEY

Indicates the original value of the changed DBKEY, or when data is journaled: hexadecimal representation of bytes 17 through 23 of the record. RL-ALT-IMAGE-CHNGD-DBKEY is located in column 243 of the //SUPCRECL file.

This field contains a DBKEY when the updated record was not a target of a DML verb and only pointers were journaled. The DBKEY is the original value of the Pointer that was changed. When the verb is a CONNECT or a DISCONNECT and the record involved is not one of the adjacent records in the SET but rather the "target" of the DML-verb—then this will always contain the value CONCT-OR-DISCON.

When the verb is one that causes data to be journaled, then this field contains the hexadecimal representation of the bytes 17 through 23 of the data portion of the journaled record.

RL-DDNAME

Indicates the DNA where the record is physically stored. RL-DDNAME is located in column 259 of the //SUPCRECL file.

This contains the DDNAME (derived from the DMCL information in the //DBTRECIN dataset) associated with the file where this record is stored physically on the IDMS database.

RL-ENVIRONMENT

Indicates the one-byte code signifying the environment the program that accessed this record was running. RL-ENVIRONMENT is located in column 268 of the //SUPCRECL file. These are the possible values:

Value	Description
D	Online (short for DBDC)
B	Batch
C	CICS (application program loaded in CICS address space)

RL-SCENARIO

Indicates the scenario-name or the Schema Name. RL-SCENARIO is located in column 270 of the //SUPCRECL file.

This will eventually be used to hold a name that the USER can specify on a task in the IDMS online environment prior to doing some updates that the user later wants to categorize.

There will be a SCENARIO task where the on-line user will specify an eight character scenario name identifier, which will be associated with all subsequent updates on that terminal until the task to end the SCENARIO session (ENDSCEN) is invoked. Later, ASG-TRACER may do selection of updates from that journal based on the SCENARIO ID.

Until this feature is implemented, the value in this field is the SCHEMA-name in which the record-type is defined.

RL-CV-NUMBER

Indicates the CV-NUMBER of the IDMS Central Version that produced the journal archive tape. RL-CV-NUMBER is located in column 279 of the //SUPCRECL file.

This is the CV-NUMBER assigned in the SYSGEN to the IDMS Central Version which produced the journal archive file.

Note: _____

On earlier versions of IDMS, this value is not placed in the IDMS journal so it will not be correct here.

_____**RL-TASK-ID**

Indicates either Task-ID number or the unique recovery-unit number ASG-TRACER associated with the recovery-unit in which this record was updated. RL-TASK-ID is located in column 283 of the //SUPCRECL file.

Depending on which parameter setting you use when running ASG-TRACER against the journal, this number can be one of two values:

- If the parameter setting:
`SEL-TASK-NBR-OR-UNIQUE-REC T T=PLACE TASK-NUMBER IN //DBTRFMTJ`
is used, then this field contains the number of the Task through which the record was accessed.
- If the parameter setting:
`SEL-TASK-NBR-OR-UNIQUE-REC U T=UNIQUE RECOVERY-UNIT //DBTRFMTJ`
is used, then this field contains the unique number that ASG-TRACER generated. This number is applicable to the recovery-unit through which this record was accessed. In cases where a COMMIT verb is used, there may be more than one recovery unit within a given run-unit.

RL-SUBSCHEMA

Indicates the subschema that was used by this run unit. RL-SUBSCHEMA is located in column 291 of the //SUPCRECL file.

When the ASG-TRACER exit-14 is installed, this is the Subschema that was used by the run unit through which this record was accessed.

RL-COMMENT-IF-ASTERISK

Marks a line that is comment in nature. RL-COMMENT-IF-ASTERISK is located in column 300 of the //SUPCRECL file.

This will either contain an asterisk or a space. The asterisk indicates that it is a comment line used to indicate the title lines. These lines provide headers to help someone who is browsing the file to know what the various columns contain.

Appendix C

//SUPCFLDL File

Here are the fields that are in the file that ASG-TRACER produces for the baseline and the test version runs.

This file is comma-delimited and all fields are DISPLAY. This allows you to download these files to a PC if you want to perform other functions with the data.

PROGRAM-NAME

Indicates the name of the program that did the update. PROGRAM-NAME is located in column 1 of the //SUPCFLDL file.

VERB-NAME

Indicates the type of update verb that did the update. VERB-NAME is located in column 10 of the //SUPCFLDL file. The abbreviation for the VERB-NAME can be one of these options:

Abbreviation	Type of Update Verb
MO	MODIFY
ST	STORE
ER	ERASE
EP	ERASE PERMANENT
ES	ERASE SELECTIVE
EA	ERASE ALL
DI	DISCONNECT
CO	CONNECT

RECORD-NAME

Indicates the name of the RECORD that was updated. RECORD-NAME is located in column 13 of the //SUPCFLDL file.

ELEMENT-NAME

Indicates the name of the RECORD-ELEMENT (FIELD) that was updated, (from the SCHEMA description—same as you would see on IDMSRPTS printout with RECDES option). ELEMENT-NAME is located in column 30 of the //SUPCFLDL file.

Note: _____

When the update verb was either a CONNECT or a DISCONNECT, this file does not contain any information since there are no field names involved. The //SUPCRECL file does contain information associated with a CONNECT or DISCONNECT verb.

ELEMENT-VALUE

Indicates the value of the RECORD-ELEMENT (FIELD) that was changed. ELEMENT-VALUE is located in column 63 of the //SUPCFLDL file.

ELEMENT-OCCURS-NBR

Indicates the number of the occurs when the Field is an occurs field; it is enclosed in parentheses. ELEMENT-OCCURS-NBR is located in column 104 of the //SUPCFLDL file.

ELEMENT-PORITION-NBR

Indicates the part of the field (since only 40-byte "chunks" of the field are on each record). When the field is more than 40 bytes it will have several records in this file. This is the chunk number. ELEMENT-PORITION-NBR is located in column 112 of the //SUPCFLDL file.

JRNL-REC-TYPE

Indicates the image from the journal associated with one of these entries:

Entry	Description
B	Before the update
A	After the update

JRNL-REC-TYPE is located in column 116 of the //SUPCFLDL file.

Note: _____

The comparison utility only compares fields to this point in the publication. The following fields are not compared. These additional fields help you to identify the differences between the baseline and test-run versions of the programs you are using. They can also be used for customers who choose to download the how-changed-files to a different processing platform (e.g., a PC network).

USERID

Indicates the USERID of the person who initiated the update program [only present if you have installed the ASG-TRACER exit(s) on the CV that produced the journal archive tape]. USERID is located in column 118 of the //SUPCFLDL file.

RUN-UNIT

Indicates the Run-Unit number for the recovery-unit that did the update. RUN-UNIT is located in column 127 of the //SUPCFLDL file.

DBK-PAGE-NUMBER

Indicates the Page-Number part of the DBKEY. DBK-PAGE-NUMBER is located in column 138 of the //SUPCFLDL file.

LINE-INDEX-NUMBER

Indicates the Line-Index part of the DBKEY. LINE-INDEX-NUMBER is located in column 149 of the //SUPCFLDL file.

PAGE-GROUP

Indicates the page-group associated with the run unit that did the update. PAGE-GROUP is located in column 155 of the //SUPCFLDL file.

Note:

By using the page-group and the page-number, you can determine the DDNAME (file) where the record resides, by looking in your global DMCL.

SEQ

Indicates the Journal Sequence number (ASG-TRACER internal format). This is the value in the journal sequence number in a size that will fit in a single full word. SEQ is located in column 160 of the //SUPCFLDL file.

DML-VERB-NBR

Indicates the DML number. This number is increased when each new update DML verb number occurs within a run unit. DML-VERB-NBR is located in column 171 of the //SUPCFLDL file.

SCENARIO

(Not implemented yet.) When available, you will be able to set a SCENARIO ID for a series of online updates at the terminal that issues a command like DCUF SET SCENARIO ID TO XXXXXXXX. SCENARIO is located in column 175 of the //SUPCFLDL file.

CV-NUMBER

(Not implemented yet.) When available, will be the number of the CV that did the update (produced the journal). CV-NUMBER is located in column 184 of the //SUPCFLDL file.

ENVIRONMENT

Indicates the environment the update program was running in (i.e, was the program loaded in the IDMS-CV address space or another). ENVIRONMENT is located in column 188 of the //SUPCFLDL file and can be one of these codes:

Code	Description
D	Online in IDMS-CV (short for DBDC)
B	Batch job (probably using //SYSCTL)
C	CICS
T	TSO

TASK-ID

Indicates either Task-ID number or the unique recovery-unit number ASG-TRACER associated with the recovery-unit in which this record was updated. TASK-ID is located in column 190 of the //SUPCFLDL file.

Depending on which parameter setting you use when running ASG-TRACER against the journal, this number can be one of two values:

- If the parameter setting:

`SEL-TASK-NBR-OR-UNIQUE-REC T T=PLACE TASK-NUMBER IN //DBTRFMTJ`

is used, then this field contains the number of the Task through which the record was accessed.

- If the parameter setting:

`SEL-TASK-NBR-OR-UNIQUE-REC U T=UNIQUE RECOVERY-UNIT //DBTRFMTJ`

is used, then this field contains the unique number that ASG-TRACER generated. This number is applicable to the recovery-unit through which this record was accessed. In cases where a COMMIT verb is used, there may be more than one recovery unit within a given run-unit.

DATA-OR-POINTERS-ONLY-CD

Indicates the type of record from the journal, if there was data journaled, or only pointers. DATA-OR-POINTERS-ONLY-CD is located in column 200 of the //SUPCFLDL file and can contain one of these codes:

Code	Description
D	Data was journaled
P	Only pointers were journaled

Appendix D

Year 2000 Regression Testing

When ASG-TRACER with Assure is used for Year 2000 regression testing, we recommend use of another product which will accomplish date-aging against the same database that is used for baseline tests of the original non-remediated software. You should perform these basic steps:

1. Run existing non-remediated programs against a copy of the database that has recently been backed up for a baseline run.
2. Offload the journal tapes produced during the baseline run using the CA-IDMS journal archive process.
3. Restore the backup of the database so that it looks just like it did before the baseline run.
4. Run a program (available from other vendors) that will age all dates. This will add a certain number of days to each date in the database prior to running the changed programs in a simulated post-1999 date environment.
5. Execute the same jobstream that was used in the baseline run with the difference being that now you will use the remediated code and you are either running in a machine that has its date set past 1999 or using a date-simulator software (also available from other vendors), to make it appear to the remediated programs as if they are running in a post-1999 environment.
6. Offload the journal tapes produced during the remediated run again using the CA-IDMS journal archive process.
7. Run ASG-TRACER with Assure against the journal archive file(s) produced by the baseline run to produce the baseline files.
8. Run ASG-TRACER with Assure against the journal archive file(s) produced by the remediated run to produce the remediated test version files.
9. Run the site-chosen compare program (e.g., IBM's SUPERC) to identify the differences between what was updated in the baseline run versus the changes in the remediated run.

10. If the only changed data fields that are different are fields which contain date fields, we can feel more "assured" that no unintended program functionality changes (bugs) have been introduced into the code during the remediation process.

Note:

When ASG-TRACER with Assure is run to produce the how-changed files, you may optionally identify the name of each data-field that contains a date and instruct ASG-TRACER with Assure not to include these fields in the files to be compared in the //SUPCFLDL files. This reduces the mismatches identified by the compare program. This may be done by using the //FLDINCL parameter file in ASG-TRACER JCL along with the parameter setting:

```
TRKR-YN-USE-FLDINCL
```

specified in the //DBTINPUT parameter file. Specify Y for this parameter.

The how-changed files produced by Assure contain detail record occurrences that include the name and value of each data-field changed by both the baseline and the remediated runs. It is clear by reading the output of the compare program what data is being changed in a different manner by the remediated code.

There are three files that can be used as input to the change program (SUPERCL). Each contains a different level of detail information about the way the site-written programs change the data.

The file that has the fewest number of records contains only summary information (//TCRSUMRY).

Next in expected number of records is the RECORD-Level file (//SUPCRECL). This file contains two record occurrences per physical database record that was changed (one for the before value and one for the after value of the record).

The file that has the most record occurrences is the FIELD-Level file (//SUPCFLDL). When the updating verb is a MODIFY, this file contains two record occurrences for each FIELD that was changed on the database (one for the value of the field before the change, and one for the value of the field after it was changed).

Index

Symbols

//SUPCFLDL file

CV-NUMBER 41
DATA-OR-POINTERS-ONLY-CD 4
2
DBK-PAGE-NUMBER 41
DML-VERB-NBR 41
ELEMENT-NAME 40
ELEMENT-OCCURS-NBR 40
ELEMENT-PORTION-NBR 40
ELEMENT-VALUE 40
ENVIRONMENT 41
JRNL-REC-TYPE 40
LINE-INDEX-NUMBER 41
PAGE-GROUP 41
PROGRAM-NAME 39
RECORD-NAME 39
RUN-UNIT 40
SCENARIO 41
SEQ 41
TASK-ID 42
USERID 40
VERB-NAME 39

//SUPCRECL file

JRNL-SEQNC 32
NEW-CHANGED-DBK 35
RL-1ST-16-BYTES-DISPLAY 30
RL-1ST-16-BYTES-HEXADECIMA
L 30
RL-ALT-IMAGE-CHNGD-DBKEY
36
RL-CHGD-SET-OWN-TYP-DBK 33
RL-COMMENT-IF-ASTERISK 37
RL-CV-NUMBER 36
RL-DATA-OR-POINTERS-ONLY-C
D 33
RL-DATE-OF-UPDATE-CCYYMM
DD 35
RL-DBK-PAGE-NUMBER 31
RL-DDNAME 36
RL-DML-VERB-NBR 34
RL-ENVIRONMENT 36

RL-JRNL-REC-TYPE 30
RL-LINE-INDEX-NUMBER 31
RL-MODIFY-1ST-DIFF-OFFSET 31
RL-OWNERS-DBKEY 33
RL-PAGE-GROUP 34
RL-PROGRAM-NAME 29
RL-RECORD-NAME 29
RL-RUN-UNIT 31
RL-SCENARIO 36
RL-SUBSCHEMA 37
RL-TASK-CODE 34
RL-TASK-ID 37
RL-TERMINAL 35
RL-THIS-DBKEY-4-SR3-RECORD-
CD 31
RL-TIME-OF-UPDATE-HHMMSS
MMMMMM 34
RL-TYPE-CHANGED-DBKEY 35
RL-USERID 31
RL-VERB-NAME 29
RL-VIASET-OWNER-DBKEY 32
RL-VIA-SET-OWN-TYP-DBK 32
RL-YN-TARGET-OF-DML-V-REC
34

//TCRSUMRY file

COUNT 28
PROGRAM-NAME 27
RECORD-NAME 27
SET-NAME 28
USERID 27
VERB-TYPE-OF-UPDATE 27

@TE0BREF 11-12

@TG0BRRF 11-12

B

baseline run 1

benefits

journal comparison 4

standard utilities 4

straightforward reports 4

C

- combining
 - TCRFSUMM and TCRFCOMP 22
 - TCRPSUMM and TCRPCOMP 17
- COMPREP 17
- conventions page iv

D

- DASD requirements 5
- DD Names
 - //SUPCFLDL 13
 - //SUPCRECL 13
 - //TCRSUMRY 13

F

- FIELD-LEVEL file 2
- file requirements 5
- files
 - FIELD-LEVEL 2
 - RECORD-LEVEL 1
 - SUMMARY 1

H

- hardware requirements 5
- how-changed files 3

I

- IEBGENER 6
- installation 6
 - customizing JCL 7
 - files
 - reading 7
 - unloading 7

P

- parameters
 - //DBTINPUT 12
 - batch journal manager 9

R

- RECORD-LEVEL file 1
- regression testing 43
- reports
 - comparison summary 16
 - element comparison 22
 - element summary 21
 - processing summary 15
 - TCRFCOMP 22
 - TCRFSUMM 21
 - TCRPCOMP 16
 - TCRPSUMM 15
- requirements
 - DASD 5

- file 5
- hardware 5
- software 5

- runs
 - baseline 1

S

- software requirements 5
- SORTCARD file 17
- SUMMARY file 1
- SUPERC 1

T

- TCR.ASR.JCLLIB 6
- TCR.ASR.LOADLIB 6
- TCR.ASR.UNLOAD 6
- TCRFCOMP
 - report 22
- TCRFSUMM
 - report 21
- TCRPCOMP
 - report 16
- TCRPSUMM
 - report 15
- test version run 1, 13

U

- unloading installation files 7

Y

- year 2000
 - regression testing 43

